Pitapat Campus
Requirements and Specification Document
Rev. 1.0, 10/12/2022

Project Abstract

Pitapat Campus is a service setting up dates for college students on the same campus. Any college students trying to find a partner on their campus can use the service. Compared with other dating services, Pitapat Campus checks users' registrations through email authentication and allows users to explore candidates only in the same university.

Users should sign up and log in to use the features of the services. Email authentication is required to create an account; a user proves one's registration by providing a college email address (ex. @snu.ac.kr) and checking the verification code. After filling up basic information, including pictures, and adding tags, users can log in and access to main features: exploring profiles, exchanging "pitapat", and chatting.

Users can look at others' profiles in the search tab and send "pitapat" as an expression of affection for attractive mates. When receiving "pitapat", the user can examine the requestor's profile and accept or reject the call. After granting the request, a private chatting room between matched users is opened and the pair can exchange messages.

Since dating service is more appropriate for mobile, a user interface for mobile devices is designed priorly. There are four main views: search, request list, chatting list, and profile setting. Users can access each main view by pressing the corresponding icon in the navigation bar.

Document Revision History

* Rev 1.0, 10/12/2022 – initial version

Customer

Due to the COVID-19 pandemic, college students lost the chance to get to know people in real life. Being campus mates (a.k.a CC, Campus Couple) has become much harder, so college students started to find their mates online. As a result, the "dating" board in Everytime, an online university community service, has been vitalized.

Pitapat Campus will meet the demand to find boyfriends or girlfriends in the same college online. Students who want to be campus mates will use this software.

The customers will read this document when they want to know the overall features of the service or the purposes of the developers.

Competitive Landscape

Pitapat Campus has two types of competitors. Firstly, dating apps such as Tinder and Amanda are one of the competitor types. These universal dating applications may be attractive to customers who want to get to know people in arbitrary fields. However, they are not good options for users who want to find their partner in a specific group.

Secondly, online university community services like Everytime are often used as dating platforms. Students who want to meet people in the same university can explore self-introduction posts on Everytime's "dating" board. This method hardly allows people to consider others' appearance, one of the most important factors in choosing a potential sweetheart because most users do not upload their pictures on a public board. Furthermore, there are some irrelevant threads on the board that make the board's reputation bad.

Compared to competitive services, Pitapat Campus provides an efficient method to choose one's partner in the same college community based on users' profiles, including pictures, tags, and self-introduction. Matched users can continue communication through one-to-one chatting.

User Stories

Signing Up	
Actors	a university student who want to use Pitapat Campus
Precondition	The user is on the login page.
Trigger	The user clicks on the sign-up button.
Scenario	⁻ The user selects their own university. ⁻ The mail domain is fixed, and the user enters the email, authentication mail is sent to the corresponding email. ⁻ Authentication is completed when the user checks the authentication mail and enters the verification code. ⁻ The user sets the password, and an account is created.
Postcondition	⁻ The corresponding account is created.
Exceptions	⁻ The email already exists for the app. ⁻ The authentication mail is sent, but the user doesn't check it in time. ⁻ The password is not strong.

Acceptance Test	1.
	    given: valid email
	    when: pushes confirm button
	    then: authentication mail is sent to the corresponding email
	2.
	    given: already-existing email
	    when: pushes confirm button
	    then: alert that the email already exists, and stay on that page
	3.
	    given: authentication mail is sent
	    when: user enters the verification code
	    then: proceed to the password setting step
	4.
	    given: authentication mail is sent
	    when: it is incorrect verification code or the time is over
	    then: alert that verification code is wrong or the available time hass passed, and re-send authentication mail to guide the user sign up

Making a Profile	
Actors	a user who finished sign-up
Precondition	The user only finished email authorization.
Trigger	User's email is authorized but the user does not have profiles.
Scenario	- A user finishes email auth at the email-auth page. - The user re-enters the app after the user finishes email auth, but turns off the app before making a profile.
Postcondition	- User will have a profile.
Exceptions	- The user uploads invalid files.

Acceptance Test	1.
	    given: valid inputs
	    when: pushes submit button
	    then: a user profile is made.
	2.
	    given: some necessary fields are not filled
	    when: pushes submit button
	    then: alert the user that necessary fields are not filled
	3.
	    given: the user tries to upload a profile image
	    when: if it is not an image file
	    then: alert the user that only image files are excepted
	4.
	    given: the user does not finish making a profile.
	    when: the user tries to terminate the app before submitting.
	    then: alert the user that profile making is an essential process to using the app, and ask that the user really wants to close the app

Updating a Profile	
Actors	a user who finished sign–up
Precondition	User should have their profiles.
Trigger	The user pushes "change profile" button.
Scenario	⁻ The profile-update page looks like the user–info page, but the user can update each profile element. ⁻ Every profile element could be updated, but required fields should not be empty. ⁻ After the user finishes update and click the "confirm" button, the user profile is updated.
Postcondition	⁻ User will have an updated profile.
Exceptions	⁻ The user uploads invalid files. ⁻ The user terminates the app while updating the profile.
Acceptance Test	1. 　**given**: valid inputs 　**when**: pushes submit button 　**then**: a user profile is updated 2. 　**given**: some necessary fields are updated as empty value 　**when**: pushes submit button 　**then**: alert the user that necessary fields are not filled 3. 　**given**: the user tries to upload a profile image 　**when**: if it is not an image file 　**then**: alert the user that only image files are excepted 4. 　**given**: the user does not finish updating a profile 　**when**: the user tries to terminate the app before confirming 　**then**: alert the user that the profile will not be changed if leave, and asks the user really wants to close the app

Sending Pitapat to Others	
Actors	a user who find out other impressive users
Precondition	"Pitapat" sender and receiver are not yet matched.
Trigger	The user presses "pitapat" button on the profile page.
Scenario	- The user search page contains other users' basic information such as a nickname, main profile image, and age. - The user can see the profile detail of the others by clicking the others' main profile image. - The user sends a request by pressing the pitapat button.
Postcondition	- The receiver gets a pitapat alarm, and the receiver can check it on the list of Pitapats page. - The sender's pitapat button became already-sent state.
Exceptions	

Acceptance Test	1.
	  given: a receiver didn't block a sender
	  when: the sender pushes pitapat button
	  then: pitapat alram is sent to the receiver

Seeing the List of Pitapats from Others	
Actors	a user who gets a pitapat alarm
Precondition	A sender has sent a pitapat to other user.
Trigger	The user presses a pitapat icon on the navigation bar.
Scenario	- The list of Pitapats page contains the sender's basic information such as a nickname, main profile image, and age. - The user can see the profile detail of the sender by clicking the sender's main profile image. - The user can press an accept button, which is under the sender's profile image, or refuse the request by pressing "Nah" button.
Postcondition	- After matching up, they both can see others on the chat room list and the pitapat element should be disappeared from the list. - After pressing "Nah", the pitapat element disappears from the list, and there is no additional chat room
Exceptions	- The user is matched with nobody.

Acceptance Test	1.
	    given: after receiving Pitapats from others
	    when: when the user check a received Pitapat and press receive button to exchange Pitapats
	    then: both users get matched up successfully and they both can see others on chat room list page
	2.
	    given: after receiving Pitapats from others
	    when: when the user check a received Pitapat and press Nah button to refuse
	    then: there is no matching up

	Entering the Chat Room List Page
Actors	a user who has completed profile information
Precondition	The user has signed in.
Trigger	The user presses a "chatting" icon on the navigation bar.
Scenario	- The page contains a list of chat rooms with matched users. - Each chatting list element shows matched user's name, profile picture, and recent message with him or her. - When pressing the element, the user can move into an 1:1 chatting page with that user. - When pressing the profile picture, the user can see the profile of him or her. - When sliding the element left, the "exit" button appears. The user can exit and delete chatting when pressing this button.
Postcondition	- Pressing element of some matched user in the list: move into an 1:1 chat room view with him or her. - Pressing profile picture of some matched user: move into the profile view of him or her. - Sliding the element left and pressing the "exit" button: matching is broken and the chatting room is deleted.
Exceptions	The user is matched with nobody.

Acceptance Test	1.
	    given: the user is in chat room list page
	    when: the user presses a chat room element
	    then: 1:1 chat room with him/her appears
	2.
	    given: the user is in the chat room list page
	    when: the user presses a profile picture of matched user
	    then: a profile view of him or her appears
	3.
	    given: the user is in the chat room list page
	    when: the user slides the chat room element left
	    then: the "exit" button appears at the right of the element
	4.
	    given: the user activates exit button of some element
	    when: the user pushes the exit button
	    then: the matching is broken and the chat room is deleted
	5.
	    given: the user is in the chat room list page
	    when: the user receives a message from some matched user
	    then: a badge notification appears at the right side of the chat room element of him or her

1:1 Chatting with Matched User	
Actors	a user who is matched with some other user
Precondition	Two users are matched through exchanging pitapat.
Trigger	The user presses an element in the chat room list.
Scenario	- At the upper part of the view, the name and profile picture of the counterpart is shown. When pressing the name or the profile picture, the user can see the profile or him or her. - The user can see sent or received messages with the counterpart at the center of the view. A speech bubble with messages points right if it contains a sent message, and points left with a received message. Message dispatch time is shown by the side of each speech bubble.
Postcondition	
Exceptions	- The user tries to send an empty message.

Acceptance Test	1.
	    given: the message input field contains a non-empty message
	    when: the user presses the send button
	    then: the message is sent and a speech bubble containing the message is added at the bottom of messages
	2.
	    given: the user is in the 1:1 chat room
	    when: the message input field is empty
	    then: the button is disabled
	3.
	    given: the user is in the 1:1 chat room with a matched user
	    when: the user receives a message from the counterpart
	    then: a speech bubble containing the message is added at the bottom of messages
	4.
	    given: the user is not in the 1:1 chat room
	    when: the user receives a message from some matched user
	    then: a push notification arrives

User Story Development Plan

- Signing up – Sprint 4
- Making a profile – Sprint 2
- Updating a profile – Sprint 4
- Sending pitapat to others – Sprint 2
- Seeing the list of pitapats from others – Sprint 2
- Entering the chat room list page – Sprint 3
- 1:1 chatting with matched user – Sprint 3

User Interface Requirements

– Signing up with email verification and creating a profile

login page

두근두근
캠퍼스

이메일

비밀번호

로그인

회원가입

sign up page – selecting univ and email ...

소속 대학과
학교 이메일을 입력해주세요

서울대학교

snu.ac.kr

확인

sign up page – email verification code

인증 코드가 발송되었습니다!
이메일을 확인해주세요.

인증코드   **3:00**

재전송

확인

profile making page – basic info

인증 완료!
다른 친구들을 만나기 위해
본인을 소개해보세요!!

닉네임

닉네임

생년월일

YYYY-MM-DD

단과대

공과대학

학과

컴퓨터공학부

성별

남자

관심성별

여자

다음

profile making page – tag

나를 표현하는 태그를 입력해보세요!
ex) 취미 (등산, 그림, …)

영화   맛집탐방   테니스

다음

profile making page – self intro

인증 완료!
다른 친구들을 만나기 위해
본인을 소개해보세요!!

소개글

소개글을 작성해주세요!

다음

profile making page – photo

인증 완료!
다른 친구들을 만나기 위해
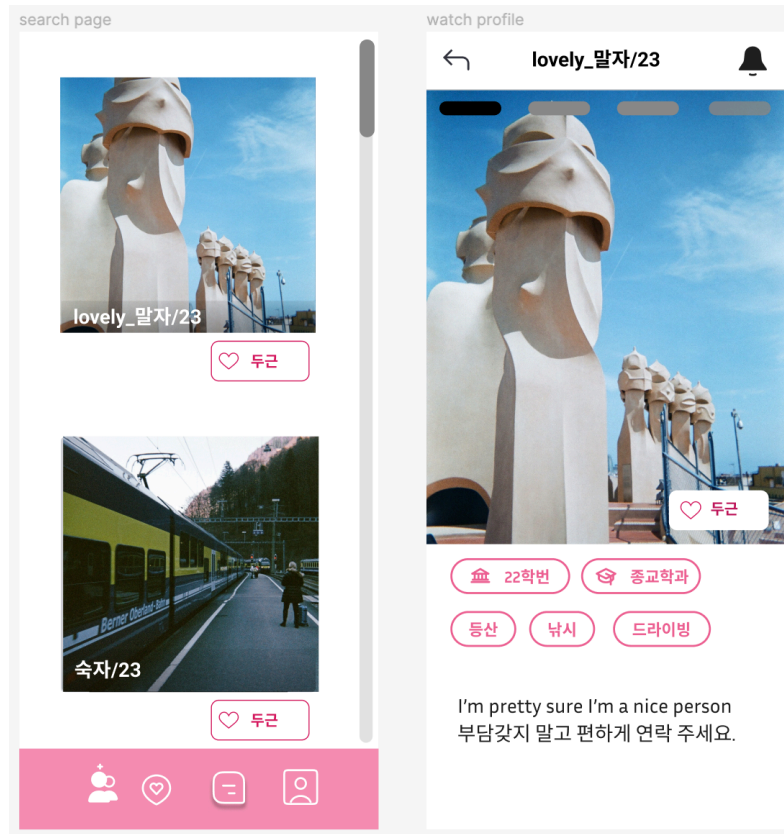본인을 소개해보세요!!

프로필 이미지

+

경고 문구
ex) 본인의 사진이어야 합니다.
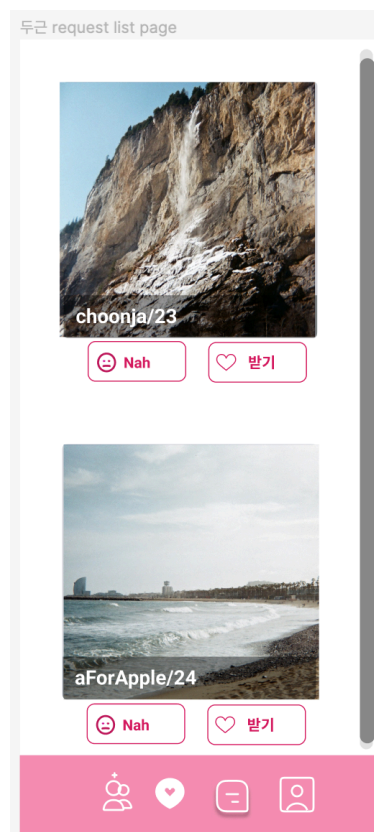
완료

profile making page – completed

프로필 작성 완료!
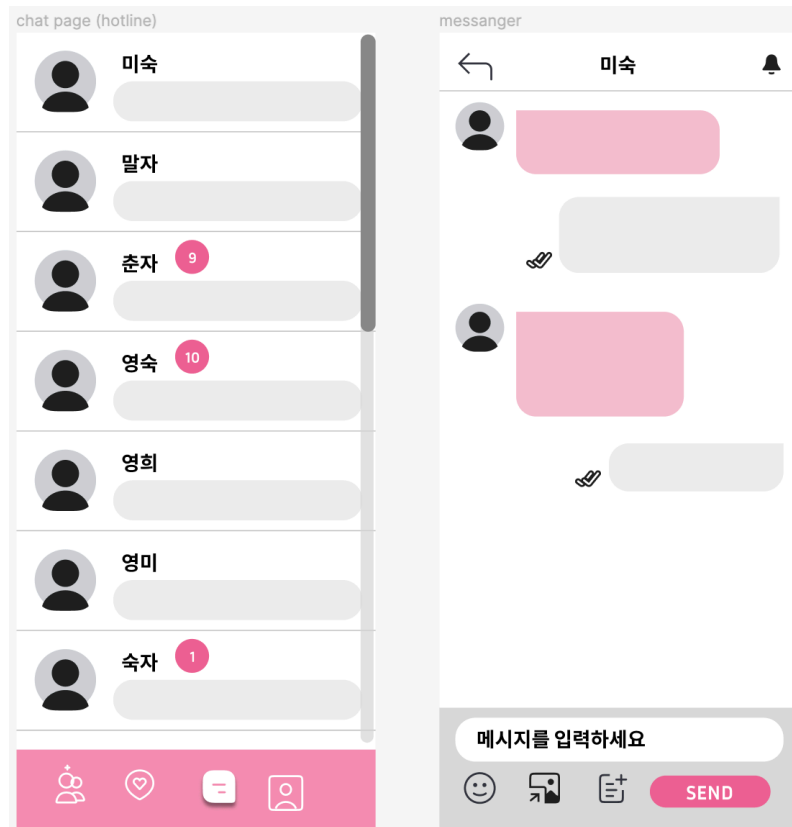다른 사람들의 프로필을 구경하고
마음에 드는 상대에게 '두근'을 보내보세요

완료

- Search page and profile detail page

-list of Pitapats page

- Chat room list page and chat room page

미숙

말자

춘자 9

영숙 10

영희

영미

숙자 1

messanger

미숙

메시지를 입력하세요

SEND

- User info page

profile setting

이용약관   ME ♂   ⚙

사진 수정

🏛 21학번   🎓 경영학과   수정

영화   맛집탐방   테니스   수정

-ENTJ
-설입 근처 거주
-부산 출신
-맛있는 음식점을 찾아다니는 것을 좋아해요
수정